

Description
SYSTEMS AND METHODS FOR PERFORMING PROTOCOL
CONVERSIONS IN A WORK MACHINE

Cross Reference to Related Applications

[01] This application claims the benefit of U.S. Provisional Application Serial No. 60/483,915 entitled "Systems and Methods for Interfacing Off-Board and On-Board Networks in a Work Machine," filed July 2, 2003, owned by the assignee of this application and expressly incorporated herein by reference in its entirety.

[02] This application is related to U.S. Application No. _____, entitled "SYSTEMS AND METHODS FOR PROVIDING SERVER OPERATIONS IN A WORK MACHINE," filed August 25, 2003, U.S. Application No. _____, entitled "SYSTEMS AND METHODS FOR PROVIDING SECURITY OPERATIONS IN A WORK MACHINE," filed August 25, 2003, U.S. Application No. _____, entitled "SYSTEMS AND METHODS FOR PROVIDING NETWORK COMMUNICATIONS BETWEEN WORK MACHINES," filed August 25, 2003, and U.S. Application No. _____, entitled "METHODS AND SYSTEMS FOR PROVIDING PROXY CONTROL FUNCTIONS IN A WORK MACHINE," filed August 25, 2003, each owned by the assignee of this application and expressly incorporated herein by reference in its entirety.

Technical Field

[03] This invention relates generally to network interface systems and more particularly, to systems and methods for performing protocol conversions in a work machine.

Background

[04] An important feature in modern work machines (e.g., fixed and mobile commercial machines, such as construction machines, fixed engine systems, marine-based machines, etc.) is the on-board electronic communications, monitoring, and control network. An on-board network includes many different modules connected to various types of communication links. These links may be proprietary and non-proprietary, such as manufacturer-based data links and communication paths based on known industry standards (e.g., J1939, RS-232, RP1210, RS-422, RS-485, MODBUS, CAN, etc.). Other features implemented with work machines are off-board networks, such as wireless networks (e.g., cellular), satellite networks (e.g., GPS), and TCP/IP-based networks.

[05] On-board modules may communicate with other on-board or off-board modules to perform various functions related to the operation of the work machine. For example, display modules may receive sensor data from an engine control module via a J1939 data link, while another control module connected to a proprietary data link may provide data to another module connected to the same link. Also, an on-board module may send data to an off-board system using a different communication path extending from the work machine to the off-board system.

[06] Problems arise, however, when modules connected to different types of data links need to communicate. These problems become especially acute as the number of data links and protocols on a given work machine increases. For example, communicating information from various protocols (e.g., J1939, RS232, RP1210, RS-422, RS-485, MODBUS, CAN, ISO11783, ATA, etc.) over a TCP/IP-based network may be problematical in current work machine environments. In addition, communicating information from multiple and different data links over a single data link protocol may pose particular difficulties. Similarly, difficulties may be presented when modules connected to

multiple and different data links simultaneously require information from a single disparate physical layer. Further, problems arise when legacy systems need to communicate with other, perhaps newer, systems that are not compatible with the legacy protocols.

[07] In certain instances, an inability to translate and communicate among and between protocols may impose significant limitations on the design and configuration of work machines and modules. For example, the placement of modules within a given work machine or environment may be limited to the range of a particular protocol. Maritime-specific modules, for instance, may only be placed within a relatively short distance of the vessel's engine due to the limitations of a J1939 data link. In certain applications, an Engine Control Module (ECM) connected to a J1939 link may need to communicate with a J1939 display module. The distance at which the display can be located from the ECM is therefore dictated by the J1939 data link range. This constraint may impose limitations not only on the placement of the two modules, but on other systems and modules in the work machine or environment.

[08] To address these problems, conventional systems may incorporate various interface devices to facilitate communications between different types of data links. Although this solution may be functionally acceptable in some instances, their implementations are restricted due to the hardware and service capabilities associated with the types of data links used in a work machine. Further, the additional hardware may take up valuable space needed for other components used by the machine.

[09] U.S. Patent No. 5,555,498 to Berra et al. describes an interface adapter for vehicle computer systems. The adapter enables existing diagnostic tools, which operate using standard communication signals, to interact with newer on-board vehicle controllers that may use incompatible communication protocols. The adapter allows the existing diagnostic tools to operate transparent of the adapter's presence. Although Berra et al. provides a solution for

interfacing existing systems with newer controllers, it is limited to diagnostic tools that communicate with on-board vehicle controllers. Further, the system described by Berra et al. does not accommodate communicating information from multiple and different data links over a single data link protocol and communicating information from a single data link to multiple and different modules. In addition, Berra et al.'s system cannot determine when protocols are inconsistent and provide corresponding interface services. Also, the system does not address the problems associated with the limited useable range of certain protocols.

[10] Methods, systems, and articles of manufacture consistent with certain embodiments of the present invention are directed to solving one or more of the problems set forth above.

Summary of the Invention

[11] Consistent with embodiments of the present invention, methods and systems may be provided for performing protocol conversions in an environment including a work machine having one or more modules coupled to one or more data links. Such methods and systems may leverage one or more gateways to perform tunneling, translating, and bridging operations.

[12] Tunneling processes consistent with embodiments of the present invention may include receiving, by a gateway, a message from a source module in a first protocol. The gateway may encapsulate the message within transmission units of a second protocol and output the encapsulated message on a second data link using the second protocol. A destination module may receive the encapsulated message from the second data link and extract the message from second protocol transmission unit.

[13] Translating processes consistent with embodiments of the present invention may include receiving, by the gateway, a message in a first data link protocol including a parameter identifier. The gateway may match the parameter identifier with a parameter identifier included in a translation table and scale

associated parameter data using a scale factor corresponding to a second data link protocol. The gateway may then provide the scaled parameter data to a module using the second data link protocol.

[14] Consistent with embodiments of the present invention, a first gateway may receive a message, from a source module, on a first data link that uses a first protocol. The gateway may encapsulate the received message within a transmission unit consistent with a second protocol. The gateway could, alternatively, translate the received message into a comparable message of the second protocol. The encapsulated (or translated) message may be broadcasted on a second data link using the second protocol and received by a second gateway. The second gateway may extract the message from the second protocol transmission unit and route the extracted message to a destination module. The second gateway may, in addition or as an alternative to extracting, translate the message received from the second data link. In certain implementations, the destination module may be located at a distance from the source module that exceeds a transmission range of the first protocol.

Brief Description of the Drawings

[15] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several aspects of the invention and together with the description, serve to explain the principles of the invention. In the drawings:

[16] Fig. 1 is a block diagram of an exemplary system that may be configured to perform certain functions consistent with embodiments of the present invention;

[17] Fig. 2 is a block diagram of an exemplary gateway consistent with embodiments of the present invention;

[18] Fig. 3 is a block diagram of an exemplary software architecture for a gateway consistent with embodiments of the present invention;

[19] Fig. 4 is a block diagram of an exemplary on-board environment in which tunneling processes may be implemented, consistent with embodiments of the present invention;

[20] Fig. 5 is a flowchart depicting an exemplary tunneling process consistent with embodiments of the present invention;

[21] Fig. 6 is a block diagram of another exemplary environment in which tunneling processes may be implemented consistent with embodiments of the present invention;

[22] Fig. 7 is a flowchart depicting another exemplary tunneling process consistent with embodiments of the present invention;

[23] Fig. 8 is a block diagram of an exemplary off-board environment in which tunneling processes consistent with embodiments of the present invention may be implemented;

[24] Fig. 9 illustrates an exemplary translation table consistent with embodiments of the present invention;

[25] Fig. 10 is a flowchart depicting an exemplary translation process consistent with embodiments of the present invention; and

[26] Fig. 11 is a block diagram of an exemplary environment in which bridging operations may be implemented, consistent with embodiments of the present invention.

Detailed Description

[27] Reference will now be made in detail to the exemplary aspects of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Overview

[28] Fig. 1 illustrates an exemplary system 100 in which features and principles consistent with an embodiment of the present invention may be

implemented. As shown in Fig. 1, system 100 may include a work machine 105 including an on-board system 110 comprising a gateway 120 and on-board modules 125, 127. System 100 may also include one or more off-board systems 130-150. Although gateway 120 is shown as a separate element, methods and systems consistent with the present invention may allow gateway 120 to be included in one or more elements, such as on-board modules 125 and/or 127.

[29] A work machine, as used herein, refers to a fixed or mobile machine that performs some type of operation associated with a particular industry, such as mining, construction, farming, etc. and operates between or within work environments (e.g., construction site, mine site, power plant, etc.). A non-limiting example of a fixed machine includes an engine system operating in a plant, off-shore environment (e.g., off-shore drilling platform). Non-limiting examples of mobile machines include commercial machines, such as trucks, cranes, earth moving vehicles, mining vehicles, backhoes, material handling equipment, farming equipment, marine vessels, aircraft, and any type of movable machine that operates in a work environment.

[30] An on-board module, as used herein, may represent any type of component operating in work machine 105 that controls or is controlled by other components or sub-components. For example, an on-board module may be an operator display device, an Engine Control Module (ECM), a power system control module, a Global Positioning System (GPS) interface device, an attachment interface that connects one or more sub-components, and any other type of device work machine 105 may use to facilitate operations of the machine during run time or non-run time conditions (i.e., machine engine running or not running, respectively).

[31] An off-board system, as used herein, may represent a system that is located remote from work machine 105. An off-board system may be a system that connects to on-board system 110 through wireline or wireless data links. Further, an off-board system may be a computer system including known

computing components, such as one or more processors, software, display, and interface devices that operate collectively to perform one or more processes. Alternatively, or additionally, an off-board system may include one or more communications devices that facilitates the transmission of data to and from on-board system 110.

[32] Gateway 120 represents one or more interface devices configured to perform functions consistent with various embodiments of the present invention. Gateway 120 may be configured with various types of hardware and software depending on its application within a work machine. Thus, in accordance with embodiments of the invention, gateway 120 may provide interface capability that facilitates the transmission of data to and from on-board system 110, performs various data processing functions, and maintains data for use by one or more on-board modules or off-board systems. For example, gateway 120 may be configured to perform protocol conversions (e.g., tunneling and translations), intelligent routing, and server-based operations, such as data provisioning, application provisioning, Web server operations, electronic mail server operations, data traffic management, and any other type of server-based operations that enable on-board system 110 to retrieve, generate, and/or provide data with on-board and/or off-board systems. For clarity of explanation, Fig. 1 depicts gateway 120 as a distinct element. However, consistent with principles of the present invention, “gateway” functionality may be implemented via software, hardware, and/or firmware within one or more modules (e.g., 125 and/or 127) on a network, which controls a system on a work machine and communicates with an off-board system. Thus, gateway 120 may, in certain embodiments, represent functionality or logic embedded within another element.

[33] On-board module 125 represents one or more on-board modules connected to one or more proprietary data links 128 included in on-board system 110. On-board module 127 may be one or more on-board modules connected to a non-proprietary data link 129, such as Society of Automotive Engineers (SAE)

standard data links including Controller Area Network (CAN), J1939, etc.
standard data links.

[34] As shown in Fig. 1, gateway 120 also interfaces with one or more off-board systems 130-150. In one exemplary embodiment, off-board systems 130-150 include, for example, computer system 130, computer system 140, and service port system 150.

[35] Computer system 130 represents one or more computing systems each executing one or more software applications. For example, computer system 130 may be a workstation, personal digital assistant, laptop, mainframe, etc. Computer system 130 may include Web browser software that requests and receives data from a server when executed by a processor and displays content to a user operating the system. In one embodiment of the invention, computer system 130 is connected to on-board system 110 through one or more wireline based data links, such as a Local Area Network (LAN), an Extranet, and the Internet using an Ethernet connection based on TCP/IP.

[36] Computer system 140 also represents one or more computing systems each executing one or more software applications. Computer system 140 may be a workstation, personal digital assistant, laptop, mainframe, etc. Also, computer system 140 may include Web browser software that requests and receives data from a server when executed by a processor and displays content to a user operating the system. In one embodiment of the invention, computer system 140 is connected to on-board system 110 through one or more wireless based data links, such as cellular, satellite, and radio-based communication data links.

[37] Computer systems 130 and 140 may each be associated with a user (e.g., customer), multiple users, a business entity (dealer, manufacturer, vendor, etc.), a department of a business entity (e.g., service center, operations support center, logistics center, etc.), and any other type of entity that sends and/or receives information to/from on-board system 110. Further, computer

system 130 and 140 may each execute off-board software applications that download or upload information to/from on-board system 110 via gateway 120. In certain embodiments, computer systems 130 and 140 may include one or more controllers such as a PLC (Programmable Logic Controller), which could be used in plants/factories.

[38] Service system 150 represent one or more portable, or fixed, service systems that perform diagnostics and/or service operations that include receiving and sending messages to on-board system 110 via gateway 120. For example, service system 150 may be a electronic testing device that connects to on-board system 120 through an RS-232 serial data link. Using service system 150, a user or an application executed by a processor may perform diagnostics and service operations on any of on-board system modules 125, 127 through gateway 120.

[39] In one embodiment, gateway 120 may include various computing components used to perform server based services (e.g., communications services, file services, database services, etc.) for on-board system 110. Fig. 2 shows an exemplary block diagram of gateway 120 consistent with embodiments of the present invention. As shown, gateway 120 includes a digital core 202, on-board data link port components 220-1 to 220-N, and off-board data link port components 225-1 to 225-Y.

[40] Digital core 202 includes the logic and processing components used by gateway 120 to perform its interface, communications, and server functionalities. In one embodiment, digital core 202 includes one or more processors 205 and internal memories 210 and 215. Processor 205 may represent one or more microprocessors that execute software to perform the gateway features of the present invention. Memory 210 may represent one or more memory devices that temporarily store data, instructions, and executable code, or any combination thereof, used by processor 205. Memory 215 may represent one or more memory devices that store data temporarily during operation of gateway

120, such as a cache memory, register devices, buffers, queuing memory devices, and any type of memory device that maintains information. Memories 210 and 215 may be any type of memory device, such as flash memory, Static Random Access Memory (SRAM), and battery backed non-volatile memory devices.

[41] On-board data link ports 220-1 to 220-N represent one or more interface devices that interconnect one or more on-board data links with digital core 202. For example, on-board data link ports 220-1 to 220-N may connect to proprietary and non-proprietary data links 128, 129, respectively. In one embodiment, on-board data link ports 220-1 to 220-N interfaces with one or more proprietary data links, one or more CAN data links (e.g., J1939, galvanized isolated CAN data links, etc.), one or more RS-232 serial based data links (e.g., MODBUS, PPP, NMEA183, etc.), and one or more RS-485 data links. On-board data link ports 220-1 to 220-N may also include virtual (i.e., software) ports that allow a single connection to act as if there were multiple connections.

[42] Off-board data link ports 225-1 to 225-Y represent one or more interface devices that interconnect one or more off-board data links with digital core 202. For example, off-board data link ports 225-1 to 225-Y may connect gateway 120 to one or more RS-232 data links, RS-485 data links, Ethernet data links, MODBUS data links, radio data links, and/or satellite data links, etc. It is appreciated that gateway 120 may be configured to interface with any type of data link used in an on-board or off-board system network.

[43] The gateway 120 shown in Fig. 2 is exemplary and not intended to be limiting. A number of additional components may be included in gateway 120 that supplement and/or compliment the operations of digital core 202 and data link ports 220 and 225. For example, gateway 120 may also include an internal power supply, a real time clock, hour meter, sensor inputs for receiving signals from one or more sensors monitoring the operations of a work machine component, memory arrays, etc. Moreover, as mentioned above, gateway 120

may, in certain embodiments, be implemented (e.g., via logic and/or circuitry) within one or more modules coupled to a given network.

[44] In operation, digital core 202 executes program code to facilitate communications between on-board modules and/or off-board systems. In one embodiment of the present invention, memory 210 includes application and server-based software programs that allow information received through either data link ports 220 and 225 to be processed and/or transferred to the proper destination module/system in the proper format. Fig. 3 illustrates an exemplary software architecture model 300 that may be implemented by gateway 120 consistent with embodiments of the present invention.

[45] Exemplary model 300 may include hardware interface software, such as boot executable software and driver software layer 310, that drive the on-board and off-board data link ports 220 and 225 connecting the multiple types of data links to gateway 120 (e.g., Ethernet, RS-232, CAN, proprietary data links, etc.). A core hardware access layer 315 interfaces boot executable layer 310 and core software layer 330, which includes software associated with runtime operations of gateway 120. Layer 320 includes operating system software executed by processor 205, and layer 325 is a network stack level including one or more protocol stacks used to perform communication services, such as formatting data messages for specific protocols, etc. In one embodiment, model 300 may also include a Web server layer 335 that includes server software used by gateway 120 to perform Web server operations, such as HTML processing, content generation, Web page request processing, etc. Further, model 300 may also include one or more layers 340-360 representing application programs executable by gateway 120. For example, layers 340, 345 may represent server applications executed by gateway 120 to perform certain services, such as data provisioning, application management, traffic management, etc. Layers 360-1 to 360-X may represent application programs that perform operations associated with functions typically performed by certain types of on-board modules

connected to an on-board network, such as a Customer Communication Module (CCM), a communication adapter, a GPS Interface Module (GPSIM), a third party interface software, an Engine Vision Interface Module (EVIM), and a product link module.

[46] Model 300 may also include an inter-data link gateway layer 350 that includes one or more gateway applications 350-1 to 350-T, that perform protocol conversion operations for converting information associated with one type of data link to another. The conversion operations may include protocol translation and tunneling features. Processor 205 may execute a selected one of application programs 350-1 to 350-T based on the type of format required by an outgoing data link. For example, application layer 350-1 may represent a protocol conversion program that allows data messages received in a proprietary data link to be converted to a J1939 format for transmission across a J1939 data link. Other types of conversion applications may be configured in model 300 including application layers that combine one or more protocol conversion capabilities.

Protocol Conversion Operations

[47] Consistent with embodiments the present invention, methods and systems may leverage one or more gateways 120 in order to perform various protocol conversions. In these embodiments, gateway 120 may include hardware, firmware, and/or software (e.g., inter-data link gateway applications 350-1 to 350-T) for translating information among and between data link protocols and tunneling information over data link protocols. With such capabilities, gateway 120 may in certain embodiments operate as a bridge, access point, and/or repeater, extending the useable range of data links between various devices.

Tunneling

[48] Consistent with embodiments of the present invention, methods and systems may facilitate information exchange among and between a plurality of different data link protocols. In such embodiments, methods and systems may leverage one or more gateways 120 to perform protocol tunneling. As used herein, the term “tunneling” refers to encapsulating information from one data link protocol within transmission units associated with another data link protocol. Tunneling may therefore enable data of one data link to be transmitted over a different physical layer. Tunneling operations may, in certain embodiments, enable modules operating with a first protocol to communicate via one or more data links incompatible with the first protocol. In certain embodiments, gateway 120 may include hardware, firmware, and/or software for performing tunneling processes. For example, as described above, model 300 may include an inter-data link gateway layer 350, including one or more gateway applications 350-1 to 350-T that perform tunneling processes.

[49] As mentioned above, gateway 120 may encapsulate messages of a first protocol within a message consistent with one or more other data link protocols. Non-limiting examples of protocols with which tunneling may be used include: J1939, ISO11783, RS-422, ATA, CANOpen, TCP/IP, Ethernet, and a proprietary data link (PDL). In one example, gateway 120 may tunnel PDL messages (i.e., the passenger protocol) over J1939 and/or Ethernet data links (i.e., the encapsulating protocol). Additionally, or alternatively, gateway 120 may encapsulate PDL, J1939, ISO11783, ATA, CANOpen, and other protocols into TCP/IP messages to communicate over the Internet. Gateway 120 may also tunnel Internet Protocol (IP) packets over a J1939 data link. In certain embodiments, gateway 120 may tunnel messages from multiple and different data links simultaneously over another single data link. Gateway 120 may also receive messages from one data link and simultaneously tunnel information out on multiple and different data links. Consistent with embodiments of the present

invention, each of application programs 350-1 to 350-T, included in gateway 120, may represent a distinct protocol conversion program. For example, application layer 350-1 may represent a protocol conversion program that allows data messages received in a PDL to be encapsulated for transmission across a J1939 data link.

[50] Consistent with embodiments of the present invention, methods and systems may perform tunneling operations in an “on-board” environment or system. That is, information may be tunneled among and between data links within a given work machine. Fig. 4 is a block diagram of an on-board system 400 associated with a work machine 405, either fixed or mobile, consistent with certain embodiments of the present invention. As shown, work machine 405 may include a gateway 410, which may be similar in configuration and operation as gateway 120 described above in connection with Figs. 1 and 2. Further, work machine 405 may include one or more on-board modules connected to one or more data links. For example, as illustrated in Fig. 4, a module 415 may be coupled to a data link 420; and modules 416a, 416b, and 416c may be coupled to data links 421a, 421b, and 421c, respectively. As illustrated, gateway 410 may be interposed between data link 420 and data links 421a-421c. Modules 415 and 416a-416c may include any type of on-board module, component, or sub-component operating within work machine 405 and may be connected to one or more proprietary and/or non-proprietary data links. For example, modules 415 and 416a-c may be ECMs, J1939 display devices (e.g., sensor gauges, etc.), EVIMs, on-board diagnostic systems, etc. The illustrated modules could also represent components and systems associated with work machine 405 such as propulsion, communications, and navigation systems. Data links 420, 421a, 421b, and 421c may be proprietary and/or non-proprietary data links similar to data links 128 and 129 described in connection with Fig. 1.

[51] Fig. 5 is a flowchart depicting an exemplary on-board tunneling process consistent with embodiments of the present invention. The illustrated

process may begin with configuring system 400 (Step 510). Configuring the system may include configuring gateway 410. In certain embodiments, gateway 410 may be configured to perform specific tunneling processes. For example, gateway 410 may be configured to utilize application layer 350-1 to tunnel messages received from a PDL across a J1939 data link. Specific tunneling applications may be selected based on the types of protocols used in a given work machine. In one example, gateway 410 may also be configured with specific encryption/decryption algorithms. In certain embodiments, configuring the system (Step 510) may include installing, connecting, configuring, and/or associating with gateway 410 modules 415 and 416a-c. For example, one or more modules may be configured to serve as tunnel exit points. That is, certain modules may be aware that messages received from gateway 410 are encapsulated, and these modules may be configured to unwrap the messages upon receipt. The configuring stage may also involve setting parameters associated with tunneling, such as timing, priority, and/or error handling parameters.

[52] Once the gateway and modules in work machine 405 are appropriately configured, data may be received in a first protocol (the passenger protocol) from one or more on-board sources (Step 520). For example, module 415 (e.g., an ECM) may send a PDL message destined for module 416a to gateway 410 over data link 420 (e.g., a PDL). In this fashion, modules 415 and 416a may serve as source and destination modules, respectively. In certain embodiments, module 415 may identify the message as a public or private message by programming an ID portion of the message with an identifier designating the message as either private or public. A public message is one that is accessible to non-proprietary systems and components, such as on-board system control module 127 shown in Fig. 1. A private message is one that is directed to components within an on-board proprietary network. For example, any message identifier within an exemplary range of values (e.g., 0-1000) may be

considered public, while identifiers having values within another range (e.g., 1001-2000) may be private.

[53] Upon receiving the message in the first (i.e., passenger) protocol from a source, gateway 410 may encapsulate the message within a transmission unit/format (the encapsulating protocol) consistent with the data link (e.g., data link 421a) to which the destination is coupled (Step 530). For example, data link 421a (coupled to module 416a) may be a J1939 data link, and gateway 410 may therefore encapsulate the received message within an J1939 transmission unit. In this fashion, gateway 410 may serve as a tunnel entry point. In certain embodiments, gateway 410 may also encrypt the message. In certain embodiments, gateway 410 may be pre-configured (e.g., at step 510) to encapsulate messages received from a particular data link or module in a specific protocol. In alternative embodiments, gateway 410 may be configured to discover that a message needs to be tunneled. That is, upon receiving a message from module 415, for example, gateway 410 may dynamically determine that the received message protocol is not compatible with the data link to which the destination module (e.g., 416a) of the message is coupled. For example, gateway 410 may learn that data links 420 and 421a are incompatible and therefore tunnel all data traffic that is received from data link 420 and destined for data link 421a. Gateway 410 may also analyze an identifier (e.g., a destination address) in a received message to determine whether it should tunnel the message in a second protocol. Once a message is encapsulated, gateway 410 may transmit the message to the destination (e.g., module 416a) via the data link coupled to the destination (e.g., data link 421a) (Step 540). In this fashion, gateway 410 may transmit the passenger protocol message across a data link via the encapsulating protocol.

[54] In the process of Fig. 5, one or more destination modules may receive tunneled messages. For example, destination module 416a may receive the encapsulated message from data link 421a. Upon receiving an encapsulated

message, the destination module may extract (i.e., unwrap) the message (Step 550). That is, the destination module may extract the passenger message from the encapsulating protocol. Module 416a may, for instance, extract PDL data from a J1939 message received from data link 421a. Consistent with principles of the present invention, the destination module may be configured to process the extracted message. For example, module 416a may extract PDL messages from J1939 transmission units and process the PDL data. In this fashion, tunneling processes consistent with the illustrated process of Fig. 5 may enable modules that operate with a first protocol to receive messages from an incompatible data link protocol.

[55] Although the process concerning Figs. 4 and 5 refers to tunneling a message from a single module to another single module, methods and systems consistent with the present invention may perform tunneling operations for multiple and different data links, both discretely and simultaneously. For example, gateway 410 may receive one or more messages in a first protocol (e.g., from data link 420) and discretely or simultaneously encapsulate the message(s) in multiple and different messages for transmission over different data links (e.g., 421a, 421b, and 421c). In addition, gateway 410 may receive messages from multiple data links (e.g., PDL, J1939, CANOpen, and ISO11783) and, discretely or simultaneously, encapsulate the messages in a single protocol (e.g., RS-422).

[56] In certain embodiments of the present invention, a plurality of gateways may be included in an on-board system to facilitate tunneling. Fig. 6 is a block diagram of an on-board system 600 associated with a work machine 605 in which multiple gateways are leveraged. As shown, work machine 605 may include a gateways 610 and 611, which may be similar in configuration and operation as gateway 120 described above in connection with Figs. 1 and 2. In the illustrated system, gateways 610 and 611 may serve as tunnel entry and exit points, respectively. Gateways 610 and 611 may be coupled via a data link 621. Further, work machine 605 may include one or more on-board modules

connected to one or more data links. For example, as illustrated, module 415 may be coupled to a data link 620, and module 416a may be coupled to a data link 622. Data links 620, 621, and 622 may be any type of proprietary and/or non-proprietary data links. In one example, work machine 605 may be a vessel, with gateway 610 and module 415 located in an engine room and gateway 611 and module 416a located in a pilot room.

[57] Fig. 7 is a flowchart depicting an exemplary on-board tunneling process in which multiple gateways are leveraged consistent with embodiments of the present invention. The illustrated process may include similar steps as described above in connection with Fig. 5. For example, steps 710-730 may parallel steps 510-530. After encapsulating a message received from a source (Step 730), however, the process of Fig. 7 may include transmitting the message from gateway 610 via the encapsulating protocol to gateway 611 via data link 621 (Step 740). Gateway 610 and 611 may be pre-configured to serve as tunnel entry and exit points such that messages received by gateway 610 from source modules are tunneled through gateway 611 to destination modules. Upon receiving the encapsulated message, gateway 611 may extract the passenger message from the encapsulating protocol (Step 750). At this stage, gateway 611 may route the message to one or more destination modules (Step 760). For example, gateway 611 may route the original message to module 416a across data link 622 (e.g., a PDL). In alternative embodiments, module 416a may be configured to unwrap the encapsulated message, and gateway 611 may simply route the encapsulated message from data link 621 to module 416a.

[58] In certain embodiments, tunneling processes implemented in system 600 may enable information exchange among a plurality of different data links. For example, data links 620, 621, and 622 may each use a different protocol. In such scenarios, gateway 610 may tunnel messages received from module 415 across data link 621 to gateway 611. Upon receiving the

encapsulated message, gateway 611 may tunnel the encapsulated message to module 416a, where the original message is extracted.

[59] In addition, or as an alternative, to on-board tunneling, methods and systems consistent with the present invention may perform off-board tunneling. Fig. 8 is a block diagram of an exemplary off-board environment consistent with embodiments of the present invention.

[60] As shown in Fig. 8, a work machine 810 may include a gateway 815, which may be configured, and operates, similarly to gateway 120 described in connection with Figs. 1 and 2. Work machine 810, which may be mobile or fixed, may also include one or more modules 811, which may be similar to modules 415 and 416a-c. Gateway 815 may execute one or more server applications that allow work machine 810 to communicate, share, and/or process information with one or more off-board elements, such as one or more other work machines 820, one or more external systems 830, and/or one or more external networks such as a Wide Area Satellite Wireless Network (WASWN) 840, a Wireless Local Area Network (WLAN) 850, a Wide-Area Terrestrial Wireless Network (WATWLN) 860, and/or a Wide Area Network (WAN) 870.

[61] WASWN 840 may be a satellite radio network that includes infrastructure allowing communications between one or more satellite devices and a remote system, such as computer system 140 described in connection with Fig. 1. WLAN 850 may be a wireless radio network including infrastructure that facilitates communications between one or more wireless radio devices and a remote system, such as computer system 140. WATWLN 860 may be a wireless network that includes infrastructure allowing communications between one or more cellular devices and a remote system (e.g., computer system 140). WAN 870 may be a network including the infrastructure that allows for Internet access, such as the World Wide Web.

[62] As illustrated, work machine 820 may include a gateway 825 that may be configured and operates similar to gateway 120. Work machine 820 may

also include one or more modules 821, which may be similar to modules 415 and 416a-c. Work machine 820 may be a mobile or fixed work machine connected to work machine 810 through a wireline or wireless data link 802. External system 830 may represent a remote system that communicates with gateway 815 through a wireless or wireline connection, such as computer system 130, computer system 140, or service port system 150.

[63] Although Fig. 8 shows work machine 820 and external system 830 connected to work machine 810 through dedicated data links, these elements may also be configured to communicate with gateway 815 through one or more of networks 840, 850, 860, and 870.

[64] In the illustrated off-board environment of Fig. 8, gateway 815 may tunnel messages from one or more modules within work machine 810 to one or more off-board elements. For example, module 811 in work machine 810 may need to transmit information to module 821 in work machine 820. In such situations, gateways 815 and 825 may be configured as tunnel entry and exit points, respectively. That is, gateway 815 may received a message destined for module 821 from module 811, encapsulate the message in a transmission unit consistent with data link 802, and transmit the encapsulated message across data link 802 to work machine 820. Upon receiving the encapsulated message, work machine 820 may route the message to gateway 825, where it is then unwrapped and routed to module 821.

[65] In certain embodiments consistent with the present invention, gateway 815 may be configured to encapsulate messages (originating from modules within work machine 810) within network packets for transmission across one or more networks. For example, gateway 815 may encapsulate PDL, J1939, CANopen, ISO11783, and other protocol data within TCP/IP messages for transmission across the Internet (e.g., via network 870). In addition, gateway 815 may be configured to tunnel IP packets across one or more data links within work machine 810. For example, gateway 815 may tunnel IP packets received from

network 870 over a J1939 and/or an ISO11783 data link coupled to module 811 within work machine 810.

[66] As mentioned above, tunneling operations may be performed for multiple and different data links, both discretely and simultaneously. Accordingly, gateway 815 may tunnel messages received from a plurality of off-board elements (e.g., network 870, work machine 820, external system 830) across one or more data links to destinations within work machine 810. In addition, gateway 815 may tunnel messages received from a plurality of different data links within work machine 810 across one or more data links coupled to an off-board element. For example, gateway 815 may encapsulate data received from PDL, J1939, CANopen, and ISO11783 data links within TCP/IP messages for transmission across the Internet (e.g., via network 870).

Translating

[67] Consistent with embodiments of the present invention, methods and systems may leverage one or more gateways 120 to perform protocol translation in order to facilitate communications between different data links, whether on-board or off-board. As used herein, the term “translating” refers to converting messages from one data link protocol into comparable messages of another protocol. In exemplary translation processes consistent with the present invention, protocol-specific parameters may be translated between different data links. Such parameters may include operational parameters, such as engine speed, injection rates, component and/or area temperatures, pressures, etc. corresponding to systems, modules and components located in a work machine environment. Further, parameters may include, or be associated with, engine diagnostic and performance parameters associated with an ECM. Parameters may also reflect commands or be used to perform certain actions. Messages from modules in a work machine environment may include one or more commands to adjust one or more parameter data values based on, for example, a requested action directed to a work machine. In one instance, a message may include a

request to increase engine speed of a particular work machine by adjusting (or requesting adjustment of) data values associated with an engine speed parameter.

[68] Translating differs from tunneling in that the messages from one data link protocol are converted into comparable messages of another protocol before they are sent. That is, messages are not encapsulated into the physical layer of another protocol, as is done with tunneling. Consistent with principles of the present invention, translating may be performed among and between any number of protocols. Messages from multiple and different data links may be discretely or simultaneously translated and sent out on a single data link. Messages may also be received from a single data link and discretely or simultaneously translated and sent out over multiple and different data links. Consistent with certain embodiments of the present invention, translation processes may also include encrypting and/or decrypting data link traffic. Non-limiting examples of translations include: (1) PDL and J1939 to MODBUS; (2) PDL to ISO11783; (3) PDL to J1939; (4) ATA to J1939; and vice versa. In certain embodiments, gateway 120 may include hardware, firmware, and/or software for performing translation processes. For example, model 300 may include an inter-data link gateway layer 350, including one or more gateway applications 350-1 to 350-T that perform translation processes.

[69] Consistent with principles of the present invention, a translation data structure, such as a translation table, that maps parameters between data links may facilitate protocol translation. Gateway 120 may access such a table in order to perform translation operations. Fig. 9 illustrates an exemplary translation table 910 consistent with embodiments of the present invention. In certain embodiments, translation table 910 may be stored in a memory device within gateway 120 and accessed using one or more processing devices, such as a memory controller or a CPU device. Alternatively, translation table 910 may reside external to gateway 120. As illustrated in Fig. 9, translation table 910 may include a plurality of parameter identifiers (PIDs) representing system parameters

associated with various data link protocols. For example, PID 1 may represent an engine speed (RPM) parameter associated with certain protocols. PID 2 may, as illustrated, represent a temperature parameter. Table 910 may include any number (R) of different PIDs.

[70] Consistent with principles of the present invention, table 910 may include one or more scaling factors, each representing a data link “view.” Each view may correspond to a particular protocol interfaced by gateway 120. Table 910 includes four views: (1) a PDL view; (2) an Ethernet data link (i.e., Web) view; (3) a J1939 view; and (4) a RS-422 view. Although four views are shown, table 910 may include any number of views corresponding to data links interfaced by gateway 120. Each “view” may enable its associated data link to interpret parameter data stored in a universal storage location. Universal Storage (US) represents a memory location or locations that store one or more values corresponding to a particular parameter (i.e., parameter data). Parameter data may be received from one or more data links interfaced by gateway 120.

[71] As illustrated in Fig. 9, each data link view may include a scale factor corresponding to translation logic used by gateway 120 to translate parameter data stored in the US to an appropriate format for the particular data link protocol. In certain embodiments, all views represented by translation table 910 may support a given parameter. For example, an RPM parameter (PID 1) may exist in all of the protocols mapped by translation table 910 (e.g., PDL, Web, J1939, and RS-422). However, in some cases, certain parameters may be supported by less than all of the views mapped by translation table 910. For example, the temperature parameter may be supported by PDL, Ethernet, and J1939 but not by RS-422. The scale factor for such non-supporting views may be null or set to zero.

[72] In addition, each view in translation table 910 may include a specific read/write privilege to the Universal Storage. That is, certain data links

may be assigned write privileges to the Universal Storage, while other data links have only read access.

[73] Consistent with translating processes of the present invention, translation table 910 may be pre-configured with a plurality of parameter identifiers and scale factors corresponding to a plurality of data links interfaced by gateway 120. In operation, gateway 120 may receive a message, including a PID and corresponding parameter data, from a particular data link. In response to such a message, gateway 120 may extract the PID and store the parameter data in the US. In addition, gateway 120 may use the PID to scale the parameter data according to the scale factors included in translation table 910, thereby creating multiple “views” of the parameter data. In one example, gateway 120 may receive a request for parameter data from a particular data link. The request may include a PID corresponding to the requested data. In response to such a request, gateway 120 may extract the PID from the request and scale the requested parameter data (previously stored in the US) using a scale factor corresponding to the extracted PID and requesting data link protocol.

[74] Fig. 10 is a flowchart depicting an exemplary translation process consistent with embodiments of the present invention. The illustrated process may begin when gateway 120 receives a message from a source (Step 1010). For example, on-board control module 125 (e.g., an ECM) may provide gateway 120 with a PDL message destined for computer system 130, which in this example includes an Internet browser. The received message may include one or more parameters and associated parameter data. The received message may also indicate one or more destinations for the parameter data included in the message. In certain embodiments, the received message may serve to transmit parameter data from a source to a destination. In one example, module 125 may send gateway 120 a message (e.g., over proprietary data link 128) including a PID corresponding to engine speed (e.g., RPMs) and corresponding parameter data representing actual engine speed, such as 100 RPMs. The parameter data may be

configured in a format consistent with the data link over which it is transmitted. For example, although the actual engine speed reported by module 125 may be 100 RPMs, the module may transmit information that is numerically (or textually, symbolically, etc.) different from the actual value, such as the value 200. Because other modules coupled to different data links may require the engine speed data, the parameter data needs to be appropriately translated for other protocols. Received messages may also reflect commands that instruct a destination system or module to perform an action, resulting in a change in the parameter data. For example, a message may be sent to an ECM to increase RPMs. In one embodiment, messages received from on-board systems may include a PID corresponding to the type of parameter that is being manipulated, sent, etc. in the message. In the above examples, the PID may reflect an RPM parameter.

[75] Upon receiving the message from the source, gateway 120 may extract the PID from the message and store the corresponding parameter data (e.g., 200) in the Universal Storage location (Step 1020). As mentioned above, methods and systems consistent with the present invention may enable multiple and different data links to access data stored in the US. In one example, translation table 910 may allow various views (e.g., Web, J1939, PDL, etc.) to access the stored RPM data.

[76] In the process of Fig. 10, gateway 120 may scale the parameter data in the received message from the source to conform with the destination protocol (Step 1030). For example, if the destination for the parameter value is a Web-based module, gateway 120 may use a scale factor from translation table 910 corresponding to the Web view. Gateway 120 may identify and select an appropriate scale factor based on the PID corresponding to the parameter data and destination protocol. For example, gateway 120 may scale RPM data for an Ethernet protocol by retrieving a scale factor that corresponds to the Web view and the RPM PID. Referring to the exemplary value depicted in Fig. 9, for the

Web the RPM parameter data may be scaled by one-half ($1/2$) in order to retrieve the actual RPM value of 100. In another example, the RPM parameter data may be scaled by ten (10) in order to provide a J1939 module with the actual parameter data. That is, 2000 may correspond to an actual RPM value of 100 in the J1939 protocol.

[77] After scaling the parameter value, gateway 120 may transmit (e.g., via a message) the scaled parameter value to its destination via the data link associated with the destination (Step 1040). For example, gateway 120 may transmit the scaled RPM parameter value to the Internet browser of system 130 via an Ethernet data link.

[78] Although the process of Fig. 10 refers to specific source and destination modules, translation processes consistent with the present invention may enable multiple and different processes associated with various data links to access (discretely and simultaneously) data from the US location. For example, gateway 120 may receive RPM data periodically from an ECM, and a plurality of other modules may periodically access the data from gateway 120 via translation table 910. In such scenarios, gateway 120 could receive a request, including a particular PID, from a data link for a parameter value corresponding to the PID. In response to such a request, gateway 120 may use the received PID to select an appropriate scale factor for the requesting data link and provide that data link with access to the parameter data from the US. The gateway may, for example, send a message to the requesting data link that includes the scaled parameter data. Further, gateway 120 could be configured to translate and transmit parameter data to several modules, perhaps periodically. Additionally, a particular view may access information and provide feedback forming a closed loop operation. In one instance, a J1939 module may receive RPM data from gateway 120 and, in response, provide a command destined for a PDL-based ECM to increase engine speed. Such a command may be routed to gateway 120, where it is translated and sent to the ECM.

[79] As described above, each data link view in translation table 910 may include its own read/write privileges to the Universal Storage. Thus, in the above example, the Web browser may not be permitted to overwrite the parameter value in the Universal Storage. To accommodate feedback from modules, translation table 910 may include multiple US locations corresponding to a given parameter and mapped to corresponding scale factors. For instance, if the Web browser receives parameter data from a first US location and then provides feedback, gateway 120 may store that feedback in a second US location associated with the parameter. The stored feedback may be then scaled to a format corresponding to the data link connected to original sending module. Gateway 120 may then send the scaled data to the on-board system component for processing. In one example, a PDL-based ECM may provide gateway 120 with fuel flow data. Gateway 120 may translate and route this parameter data to a diagnostic module via an Ethernet data link. Upon receipt, the diagnostic module may provide feedback to gateway which includes instructions to increase the fuel flow rate. This feedback may be stored in translation table 910, scaled, and transmitted back to the ECM for processing. In response to receiving the feedback, the ECM may increase the fuel flow rate in accord with the message from the diagnostic module.

Bridging

[80] Consistent with embodiments of the present invention, systems and methods may leverage protocol conversion capabilities to perform bridging operations. In certain embodiments, one or more gateway 120s may use protocol conversion processes (e.g., tunneling) to facilitate bridging. As used herein, the term “bridging” refers to connecting one data link having a first protocol to another data link that may use the same protocol. That is, a “bridge” may connect consistent physical layers. However, in alternative embodiments, tunneling and/or translating processes consistent with principles of the present invention may be leveraged to “bridge” inconsistent physical layers. Consistent

with principles of the present invention, information may be bridged from a first data link (e.g., a J1939 data link) over a second protocol (e.g., RS-422) to another data link consistent with the first (e.g., J1939). In this fashion, gateway 120 may extend the useable range of data links in a work machine or in a work machine environment. For instance, tunneling J1939 data over an RS-422 data link may allow the data to be transported over a distance that exceeds the range of the J1939 data link.

[81] Fig. 11 is a block diagram illustrating an exemplary on-board system 1100 in which bridging methods consistent with embodiments of the present invention may be implemented. As illustrated, a work machine 1105 in system 1100 may include a gateway 1120 and a gateway 1130, which may each be configured and operate similar to gateway 120 described earlier. A data link 1125 may be interposed between gateways 1120 and 1130. Data link 1125 may be any type of data link, such as an RS-422 data link. Gateway 1120 may be coupled to modules 1110 and 1114 via data links 1112 and 1113. Modules 1110 and 1114 could be any type of on-board modules. For example, module 1110 could be an engine and module 1114 could be a display device. Data link 1112 may, for example, be a J1939 data link, and data link 1113 may be a PDL. Gateway 1130 may, as illustrated, be coupled to a module 1134 via a data link 1132. Module 1134 may be a display device, and data link 1132 may be a J1939 data link. In one example, gateway 1120 and modules 1110 and 1114 may be located in an engine room while gateway 1130 and module 1134 are located in a pilot room of a marine machine or platform.

[82] In system 1100, modules 1110, 1114, and 1134 may be J1939-based modules. However, the distance between module 1110 and 1134 may exceed the range of the J1939 protocol. For example, the distance between modules 1110 and 1134 may be approximately 200 meters, while the range of the J1939 data link may be only 40 meters. To accommodate such a scenario, gateways 1120 and 1130 may be configured to serve as a distance bridge between

modules 1110 and 1114 and module 1134. Gateways 1120 and 1130 may be configured as bridge entry and exit points, respectively, and may be coupled together via data link 1125. Data link 1125 may have a range sufficient to accommodate the positioning of modules 1110, 1114, and 1134. For example, data link 1125 may be an RS-422 data link with a range of 300 meters.

[83] Referring to Fig. 11, gateway 1120 may receive J1939 messages from module 1110 and encapsulate the message within an RS-422 transmission unit. The encapsulated message may be transmitted over the data link 1125 and received by gateway 1130, which may extract the passenger message and route it to module 1134 over data link 1132. In this fashion, gateways 1120 and 1130 serve as a J1939 distance bridge. In certain embodiments, gateway 1120 may also receive messages destined for module 1134 from data link 1113. In one example, however, data links 1113 and 1132 may be incompatible. For instance, data link 1113 may be a PDL, and data link 1132 may be a J1939 data link. To accommodate such scenarios, gateway 1120 may translate the message from data link 1113 (e.g., to J1939) and then encapsulate the translated message within a transmission unit compatible with data link 1125. Gateway 1130 may receive the message from data link 1125, extract the passenger (e.g., the original PDL message translated to J1939), and route the extracted message to module 1134 via data link 1132. In alternative embodiments, gateway 1120 may encapsulate the message received from data link 1113, and gateway 1130 may perform translation in order to route the message to module 1134. In addition, encapsulated messages may be unwrapped by destination modules instead of gateway 1130. Thus, gateways 1120 and 1130 may serve as distance bridge while messages are unwrapped by destination modules.

[84] In certain embodiments, work machine 1105 may be a mobile work machine that monitors and controls multiple wireless on-board devices, such as radio transceivers. When mobile work machine 1105 moves into a certain range of second gateway-embedded work machine (e.g., mobile or fixed),

gateway 1120 may send a message to the gateway embedded in the second work machine. The gateway in the second work machine may translate the message to another protocol for extended transmissions (e.g., an RS-422 data link from a base station that the second work machine approaches).

[85] Although, in Fig. 11, gateways 120 and 1130 are located within work machine 1105, one or more gateways may be leveraged to bridge geographically dispersed modules and data links. For example, gateway 1120 may reside on one work machine while gateway 1130 resides in another remotely located work machine, system, or environment.

[86] For additional clarity of explanation, the systems 100-400, 600, 800, and 1100 are described herein with reference to the discrete functional elements illustrated in Figs. 1-4, 6, 8, and 11. However, it should be understood that the functionality of the illustrated elements and modules may overlap and/or may exist in a fewer or greater number of elements and modules. Elements of each system may, depending on the implementation, lack certain illustrated components and/or contain, or be coupled to, additional or varying components not shown. Moreover, all or part of the functionality of the illustrated elements may co-exist or be distributed among several geographically dispersed locations.

[87] In addition, the steps illustrated in the flowcharts of Figs. 5, 7, and 10 are consistent with exemplary implementations of the present invention. Further, it should be understood that the sequence of events described in Figs. 5, 7, and 10 are exemplary and not intended to be limiting. Thus, other method steps may be used, and even with the methods depicted in Figs. 5, 7, and 10, the particular order of events may vary without departing from the scope of the present invention. Moreover, certain steps may not be present and additional steps may be implemented in Figs. 5, 7, and 10. For example, in certain embodiments, translation, tunneling, and bridging processes consistent with embodiments of the present invention may include encrypting and/or decrypting

data link traffic. In addition, it should be understood that the illustrated stages may be modified with departing from the scope of the present invention.

Industrial Applicability

- [88] Consistent with embodiments of the present invention, methods and systems may facilitate information exchange in multi-protocol environments. In one example, a work machine may include a plurality of different data links and corresponding modules. Systems and methods of the present invention may enable these modules to exchange information, despite incompatibilities among their respective operating protocols. Consistent with principles of the present invention, methods and systems may leverage one or more gateways to perform protocol conversion processes, such as tunneling and translating. In addition, such processes may be exploited to provide features such as distance bridging. In certain embodiments, one or more gateways may be configured to perform one of tunneling, translating, or bridging. In other embodiments, one or more gateways may perform tunneling, translating, and bridging operations. In one example, a single gateway may perform tunneling, translating, and bridging operations for one or more interfaced data links.
- [89] In one embodiment, one or more gateways may be configured to encapsulate information from a first protocol within transmission units of a second protocol. In this fashion, systems and methods consistent with the present invention may tunnel messages originating from one or more modules and formatted in a first protocol through a second data link protocol to one or more destinations.
- [90] In certain embodiments consistent with the present invention, one or more gateways may be configured to translate parameters between incompatible protocols. Systems and methods of the present invention may convert parameters associated with a first protocol into comparable parameters associated with a second protocol. A translation table may facilitate such protocol translation processes.

[91] Consistent with certain embodiments of the present invention, systems and methods may perform tunneling and translating operations concurrently. Additionally, one or more gateways may be configured to tunnel information through one data link and translate information for another. Further, in certain embodiments, one or more gateways may encrypt and decrypt data link traffic in addition to performing tunneling, translating, and/or bridging.

[92] In some work machine environments, components may be located at substantial distances from each other. For example, on a marine vessel, an ECM may be located in an engine room and a display may be located several hundred meters away in a pilot room. In such environments, however, the protocols with which the modules operate may be unable to communicate information over such distances. That is, the distance between certain modules may exceed the range of the data link protocols with which these modules operate. To accommodate such situations, methods and systems of the present invention may, leveraging one or more embedded gateways, provide bridging operations in order to extend the useable range of certain protocols.

[93] Systems and methods consistent with the present invention may serve to adapt legacy systems and applications to new technology. For example, certain work machines may include multi-protocol operating environments but may include legacy ECMs not compatible with all of the implemented protocols. In some instances, it may be desirable to maintain these legacy systems despite their incompatibility with newer protocols. Methods and systems consistent with the present invention may enable such legacy systems to interact with other systems and applications that operate with incompatible physical layers. That is, the present invention may enable legacy systems to operate with newer systems and other protocols not compatible with the legacy protocol.

[94] The embodiments, features, aspects and principles of the present invention may be implemented in various environments and are not limited to work site environments. For example, a work machine with an embedded

gateway may perform the functions described herein in other environments, such as mobile environments between job sites, geographical locations and settings. Further, the processes disclosed herein are not inherently related to any particular system, and may be implemented by a suitable combination of electrical-based components. Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims.